The Triangle Algorithm: A Geometric Approach to Systems of Linear Equations

Thomas Gibson[1] Bahman Kalantari[2]

[1]Baylor University [2]Rutgers University

July 19, 2013

Rough Outline

- Quick Refresher of Basics
- Convex Hull Problem
- Solving Systems of Linear Equations

Introduction (Brief)

Definition

The *convex hull* of a finite point set $S \in \mathbb{R}^n$ is the set of all convex combinations of its points. The mathematical expression is:

$$conv(S) := \{\sum_{i=1}^{|S|} \alpha_i x_i : (\forall i\ \alpha_i \geq 0) \wedge \sum_{i=1}^{|S|} \alpha_i = 1\}$$

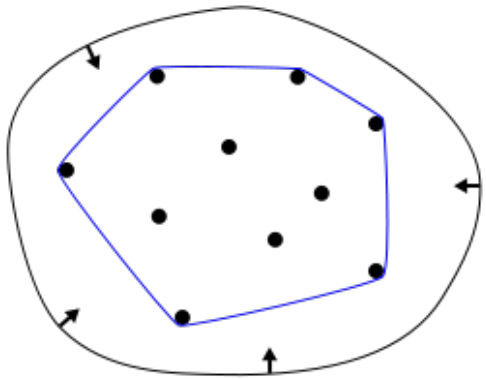
Introduction (Brief)

Definition

The *convex hull* of a finite point set $S \in \mathbb{R}^n$ is the set of all convex combinations of its points. The mathematical expression is:

$$conv(S) := \{\sum_{i=1}^{|S|} \alpha_i x_i : (\forall i\ \alpha_i \geq 0) \wedge \sum_{i=1}^{|S|} \alpha_i = 1\}$$

The convex hull of $S \in \mathbb{R}^2$ forms a *convex polygon*. And more generally, the convex hull of $S \in \mathbb{R}^n$ forms a *convex polytope*.



The Convex Hull Problem

A quick refresher: Given a point $p \in \mathbb{R}^m$, we would like to determine whether or not the point lies within the convex hull of a finite set of m-dimensional points S.

The Convex Hull Problem

A quick refresher: Given a point $p \in \mathbb{R}^m$, we would like to determine whether or not the point lies within the convex hull of a finite set of m-dimensional points S.

Theorem

(The Distance Duality Theorem) Let $S = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^m$, $p \in \mathbb{R}^m$.

1. $p \in conv(S)$ *if and only if given any* $p' \in conv(S)$, *there exists a* v_j *such that* $\| p' - v_j \| \geq \| p - v_j \|$
2. $p \notin conv(S)$ *if and only if there exists a* $p' \in conv(S)$ *such that* $\| p' - v_j \| < \| p - v_j \|, \forall\ j$.

The Convex Hull Problem

A quick refresher: Given a point $p \in \mathbb{R}^m$, we would like to determine whether or not the point lies within the convex hull of a finite set of m-dimensional points S.

Theorem

(The Distance Duality Theorem) Let $S = \{v_1, v_2, \ldots, v_n\} \subset \mathbb{R}^m$, $p \in \mathbb{R}^m$.

1. $p \in conv(S)$ *if and only if given any* $p' \in conv(S)$, *there exists a* v_j *such that* $\| p' - v_j \| \geq \| p - v_j \|$
2. $p \notin conv(S)$ *if and only if there exists a* $p' \in conv(S)$ *such that* $\| p' - v_j \| < \| p - v_j \|, \forall\ j.$

Definition

We call p' a *witness* if it satisfies $\| p' - v_i \| < \| p - v_i \|, \forall$ i.

Data: Given a vector $p \in \mathbb{R}^m$, a matrix $S \in \mathbb{R}^{mxn}$, and a fixed tolerance ϵ

Result: A vector $\overline{\alpha}$ such that $p' = S\ \overline{\alpha}$, with $\| p' - p \| < \epsilon$

initialization of p'_0 and $\overline{\alpha}_0$;

while $\| p' - p \| > \epsilon$ **do**

  for $i \leftarrow 1$ **to** *n total vertices* **do**

    if $\| p' - v_i \| \geq \| p - v_i \|$ **then**

      choose v_i to be a pivot;

    end

  end

  calculate step size $\beta = \frac{(p-p'_k)^T(v_i-p'_k)}{\|v_i-p'_k\|}$;

  update $\overline{\alpha}_{k+1} = (1-\beta)\overline{\alpha}_k + \beta\overline{\alpha}_i$;

  update $p'_{k+1} = S\overline{\alpha}_{k+1}$;

end

```
for i ← 1 to n total vertices do
    if ‖ p′ − v_i ‖ ≥ ‖ p − v_i ‖ then
        choose first v_i that satisfies condition to be a pivot;
        break;
    end
end
```

Algorithm 1: Blind Triangle Algorithm

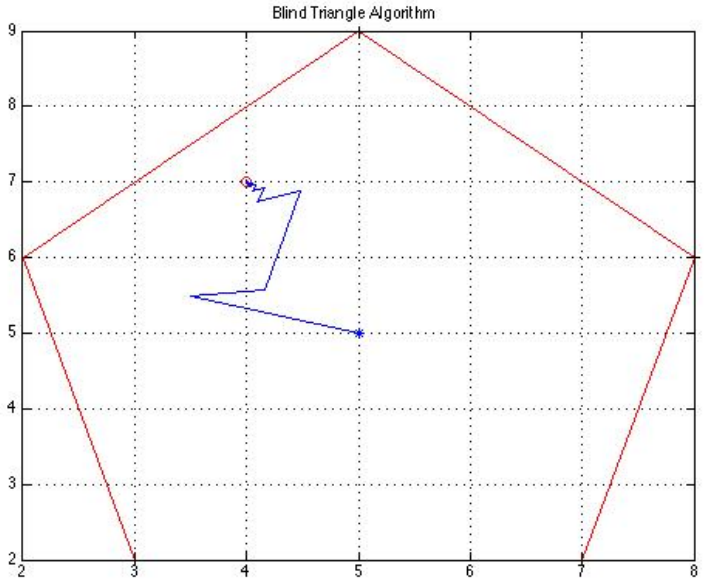


Figure : 5 Point Convex Set in 2D Space (40 iterations)

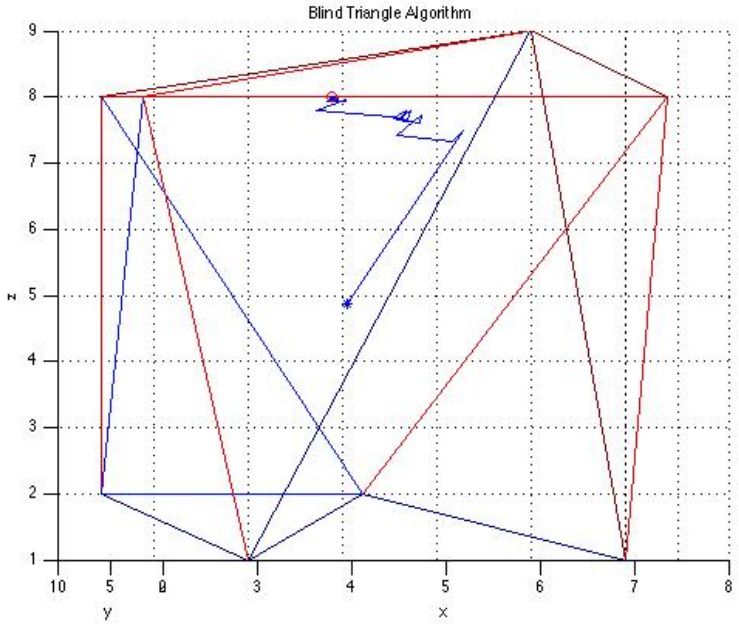


Figure : 8 Point Convex Set in 3D Space (245 iterations)

for $i \leftarrow 1$ **to** *n total vertices* **do**
  if $\| p' - v_i \| \geq \| p - v_i \|$ **then**
    find angles $\delta_i = \angle p p'_k v_i$;
  end
  find $\delta_j := \inf\{\delta_1, \delta_2, \ldots\}$;
  choose pivot v_j from corresponding angle δ_j;
end

Algorithm 2: Triangle Algorithm with Angle Checking

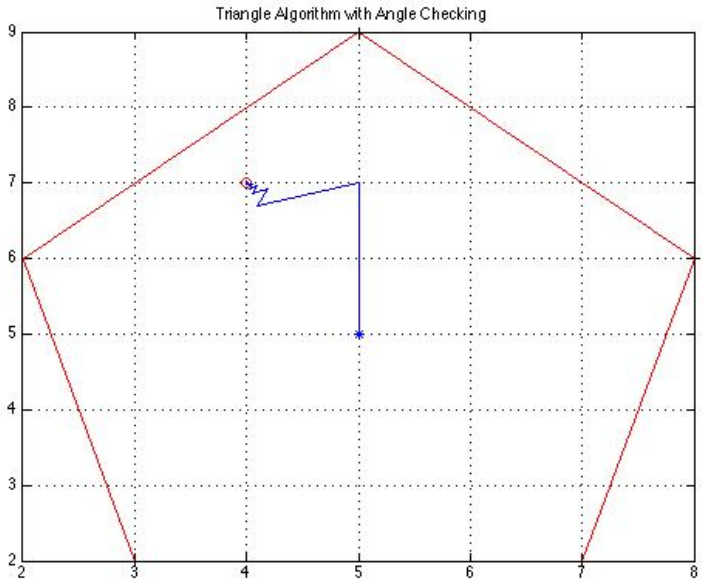


Figure : 5 Point Convex Set in 2D Space (35 iterations)

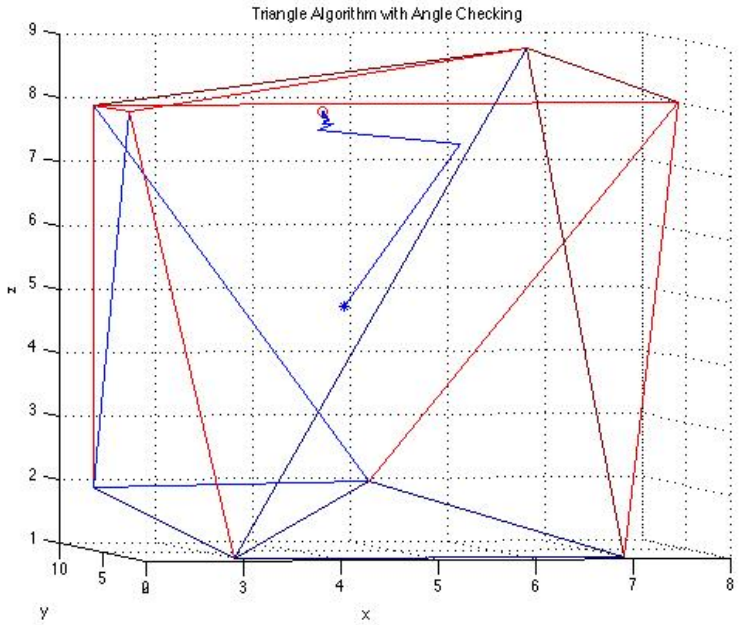


Figure : 8 Point Convex Set in 3D Space (95 iterations)

for $i \leftarrow 1$ **to** *n total vertices* **do**
  if $\| p' - v_i \| \geq \| p - v_i \|$ **then**
    find angles $\delta_i = \angle p p'_k v_i$;
  end
  find $\delta_j := \inf\{\delta_1, \delta_2, \ldots\}$;
  choose pivot v_j from corresponding angle δ_j;
end
if *two pivots* v_i *and* v_{ii} *are chosen in consecutive cycles* **then**
  increment counter;
  if *counter* $> \gamma$ **then**
    define new auxiliary point $v_{n+1} := \frac{v_i + v_{ii}}{2}$;
    $S = [S \mid v_{n+1}]$;
  end
end

Algorithm 3: Mid-Point Auxiliary Points

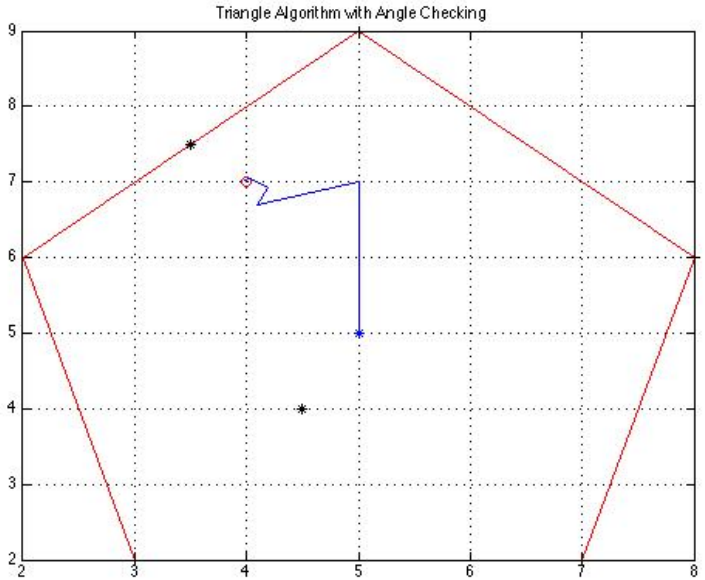


Figure : 5 Point Convex Set in 2D Space (8 iterations)

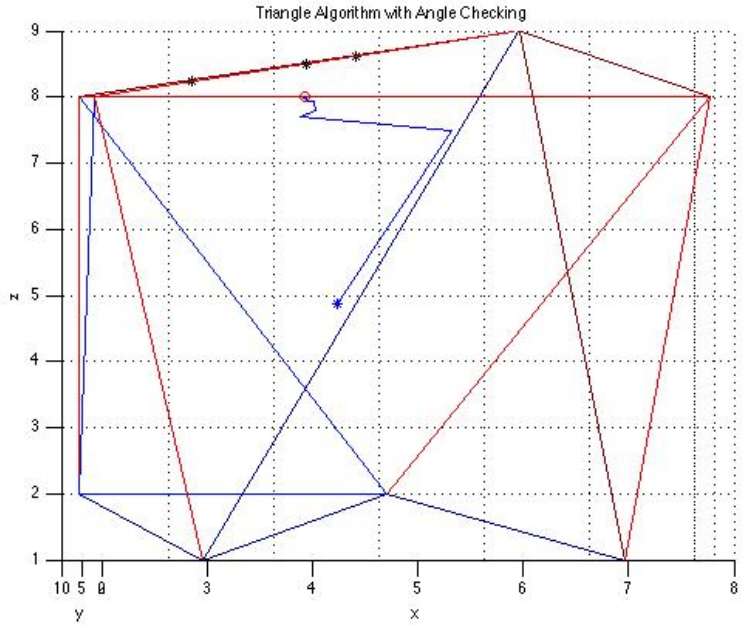


Figure : 8 Point Convex Set in 3D Space (17 iterations)

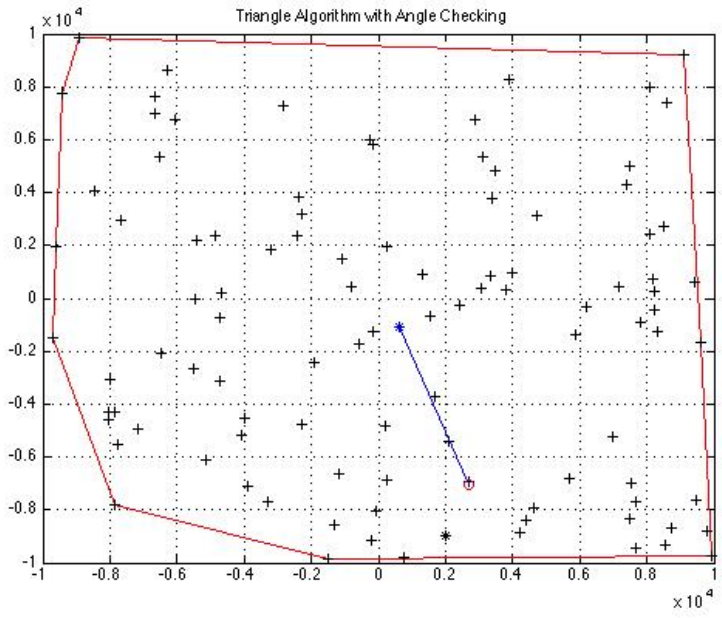


Figure : 100 Point Convex Set in 2D Space (7 iterations)

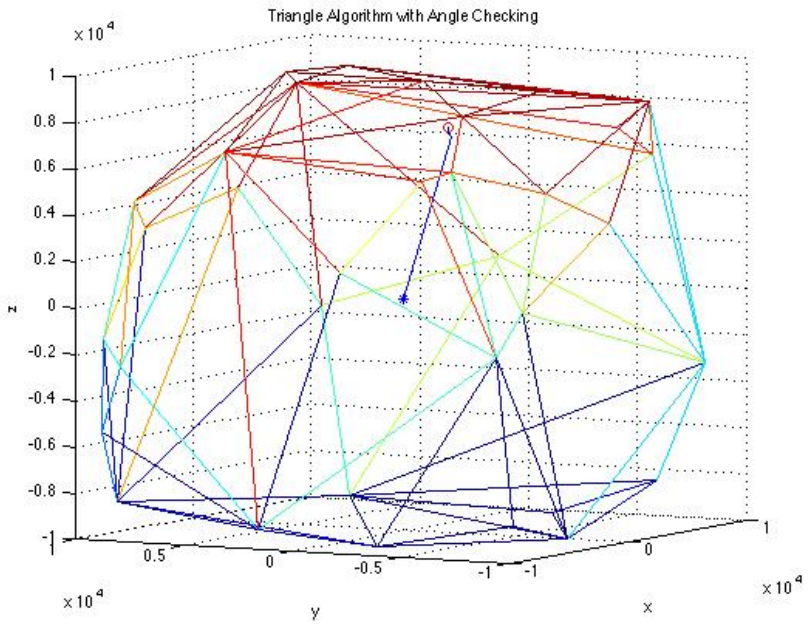


Figure : 100 Point Convex Set in 3D Space (15 iterations)

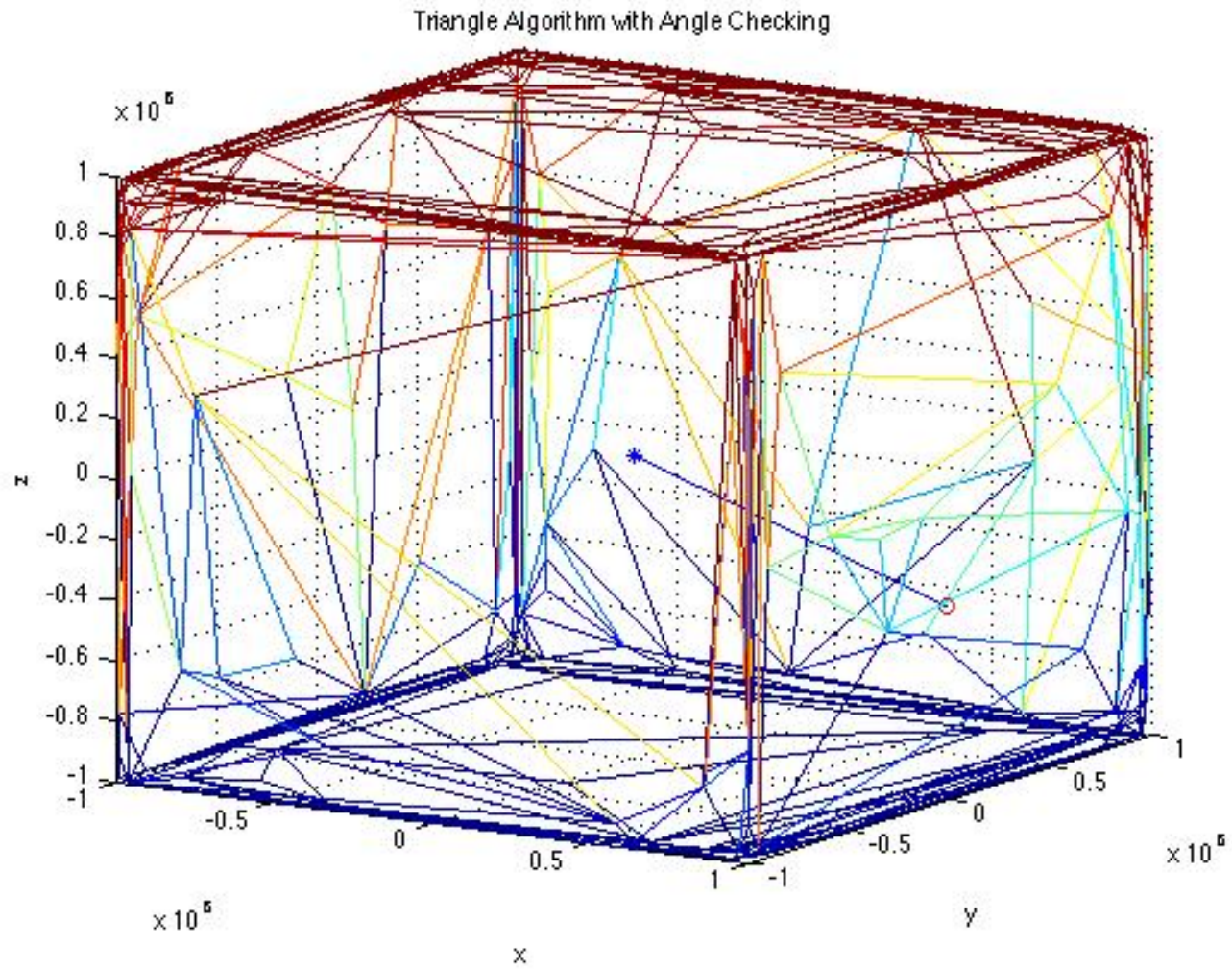


Figure : Half-Million Point Convex Set in 3D Space (5 iterations)

Solving a System of Equations
Nonnegative Case

Solving a System of Equations
Nonnegative Case

Given A $\in \mathbb{R}^{n \times n}$ and invertible, suppose $Ax = b$ has a nonnegative solution $x = A^{-1}b \geq 0$. Then we can generate an ϵ- approximate solution x_0 such that $\| Ax_0 - b \| < \epsilon$ by approximating $0 \in$ *conv([A -b])*.

Given A $\in \mathbb{R}^{n \times n}$ and invertible, suppose $Ax = b$ has a nonnegative solution $x = A^{-1}b \geq 0$. Then we can generate an ϵ- approximate solution x_0 such that $\| Ax_0 - b \| < \epsilon$ by approximating $0 \in$ *conv([A -b])*.

$$\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} & -b_1 \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} & -b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} & -b_n \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \\ \alpha_{n+1} \end{pmatrix} = \begin{pmatrix} 0_1 \\ 0_2 \\ \vdots \\ 0_n \end{pmatrix}$$

$$\sum \alpha_i = 1, \alpha_i \geq 0$$

Consider the following examples:

$$3x_1 + x_2 = 4$$
$$x_1 + 4\,x_2 + 2\,x_3 = 7$$
$$\text{-}x_2 + 3\,x_3 = 2$$

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 7 \\ 2 \end{pmatrix}$$

Solving a System of Equations
Nonnegative Case

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 4 & 2 \\ 0 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ 7 \\ 2 \end{pmatrix},$$

Comparisons	
Method	Iterations
Triangle Algorithm	1
Jacobi	11
Gauss-Seidel	7
Successive Over-Relaxation (SOR)	7

Note:

The initial guess used for Jacobi, Gauss-Seidel, and SOR was $x_0 = (\frac{4}{3}, \frac{7}{3}, \frac{2}{3})^T$. The optimal relaxation factor for SOR was approximately $\omega = 1.02$. True solution is $x_{true} = (1, 1, 1)^T$.

Solving a System of Equations
Nonnegative Case

$$\begin{pmatrix} 4 & 3 & -4 & 2 & 0 \\ 3 & 5 & 2 & 0 & 0 \\ 0 & 2 & 6 & -1 & 0 \\ 0 & -1 & -1 & 4 & 3 \\ 2 & -1 & -1 & 4 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 8 \\ 15 \\ 9 \\ 4 \\ 8 \end{pmatrix},$$

Comparisons	
Method	Iterations
Triangle Algorithm	1
Jacobi	46
Gauss-Seidel	26
Successive Over-Relaxation (SOR)	22

Note:

The initial guess used for Jacobi, Gauss-Seidel, and SOR was $x_0 = (1, 2, \frac{7}{5}, 1, \frac{9}{5})^T$. The optimal relaxation factor for SOR was approximately $\omega = 1.01$. True solution is $x_{true} = (1, 2, 1, 1, 1)^T$.

Computational Complexity (Each Iteration)	
Method	Complexity
Triangle Algorithm	$O(nm)$
Gaussian Elimination	$O(n^3)$
Jacobi	$O(n^2)$
Gauss-Seidel	$O(n^2)$
Successive Over-Relaxation (SOR)	$O(n^2)$

- Performance in high dimensions
- Comparison with Krylov Methods
- Auxilliary Point Methods

Further Work
More General System Solver

Given A $\in \mathbb{R}^{n \times n}$ and invertible, suppose $Ax = b$, but we do not know anything about x. Apply a change of variables:
Let $e = (1, 1, \ldots, 1)^T \in \mathbb{R}^m$. Then $\exists$ t ≥ 0 such that if x is a solution to:

$$A(x - te) = b$$

Given A $\in \mathbb{R}^{n \times n}$ and invertible, suppose $Ax = b$, but we do not know anything about x. Apply a change of variables:
Let $e = (1, 1, \ldots, 1)^T \in \mathbb{R}^m$. Then $\exists$ t ≥ 0 such that if x is a solution to:

$$A(x - te) = b$$

Then x ≥ 0. Define $u := Ae$, then

$$Ax = b + tu, \text{ x} \geq 0,$$

is solvable.

Data: Given $p' = A\ \alpha$ - $\alpha_{n+1}\ b \in conv(S)$
Result: A vector $\overline{\alpha}$ such that $p' = S\ \overline{\alpha}$, with $\| p' - p \| < \epsilon$
set $x_0 = \frac{\alpha}{\alpha_{n+1}}$. Define τ_0 according to:
$E(\tau_0) = \| Ax_0 - (b + \tau_0 u) \| = min\{\| Ax_0 - (b + tu) \|: t \geq t_0\}$

replace t_0 with τ_0;
if $E(t_0) \leq \epsilon_0$ **then**
  set $x_0' = (x_0 - t_0 e)$, stop;
end
Call Triangle Algorithm where $S = \{a_1, a_2, \ldots, a_n, -(b + t_0 u)\}$;
$p''(t_0) = p'' - \beta_{n+1} t_0 u$, where $p'' = A\beta - \beta_{n+1} b \in conv(S)$;
Update p' with p'' ;
Update α with β;
Update α_{n+1} with β_{n+1};

Algorithm 4: Incremental Triangle Algorithm

Further Work

- Incremental Triangle Algorithm
- Auxiliary Pivot Methods to Improve Performance
- Eigenvalue Problems

- B. Kalantari. A Characterization Theorem and an Algorithm for a Convex Hull Problem.
- B. Kalantari. Finding a Lost Treasure in Convex Hull of Points From Known Distances. 4th Canadian Conference on Computational Geometry, 2012.
- B. Kalantari. Solving Linear System of Equations Via A Convex Hull Algorithm. 29 Oct, 2012.

Thank you to all those who made my research possible:

- Bahman Kalantari
- Meng Li
- Yan Wang
- DIMACS REU Program